CONCURRENCY TEST SCENARIO GENERATION USING UML TRANSITION SEQUENCES

Concurrency is used for superior computational efficiency and/or better resource utilization. However, concurrent program execution is unpredictable due to random execution interleavings, which may give rise to different concurrency errors, namely data race, synchronization, starvation, and deadlocks.

UML model-based testing not only provides early test case design and testing effort estimation but also permits validation of the implementation. Test scenario generation from a UML design is done using UML transition sequences, which could be: (i) a state-sequence in a state machine diagram, (ii) an activity-sequence in an activity diagram, or a message sequence in a sequence diagram.

The contributions of this thesis addressing the issues as described above are: (i) New Concurrency Coverage Criteria, (ii) Concurrent Behavior Analysis of Test Scenarios, (iii) Test Scenario Generation from Non-nested Concurrent Activity Diagrams with the proposed CQS and DFS-Level-Permute algorithms, (iv) Test Scenario Generation from Nested Concurrent Activity Diagrams with Evolutionary Algorithms, and (v) Test Scenario Generation from Sequence Diagrams with Concurrency Constructs with the proposed Dependency Preserving Depth First Search (DP-DFS) algorithm.

The proposed concurrency criteria ensure the exploration of interleaving test scenarios, thereby helping the detection of concurrency errors. The proposed CQS, DFS-Level-Permute, and Evolutionary algorithms generate bug triggering scenarios and improve concurrency coverage for both non-nested and nested concurrent activity diagrams. The DP-DFS technique achieves improved concurrent message path coverage, synchronization coverage, and interleaving message path coverage from sequence diagrams. It is shown that the concurrent behavioral analysis of test scenarios generated by our approaches significantly reduces test suite size.

In a nutshell, the contributions of this thesis are primarily focused on UML transition sequence-based test scenario generation for concurrency testing. Finally, the thesis also explores the possibility of unification of both UML behavioural models – activity and sequence diagrams – through a single intermediate representation using hypergraphs. The proposed test scenario generation techniques in this thesis are compatible with UML 2.0 specifications. It is suggested that UML notations be enhanced with the addition of the extra information for concurrency testing.

* * *